Considerations in the Design of Virtual Environment Systems: A Case Study

Manuel Oliveira Jesper Mortensen Joel Jordan Anthony Steed Mel Slater

Computer Science Department

University College London

Gower Street WC1E 6PT London UK

+44 419 3876

{m.oliveira | j.mortensen | j.jordan | a.steed | m.slater}@cs.ucl.ac.uk

Abstract

There is a common understanding amongst the games industry that the Internet is a poor network infrastructure that wreaks havoc on all game engines. This view is shared amongst the Distributed Virtual Environment (DVE) research community, who traditionally find the Internet as the main culprit for disruptions in the experience of end-users.

In fact, the majority of the problems reside in the existing misconceptions regarding the nature of the Internet, leading to inefficient and inadequate network subsystems.

This paper reports the lessons learnt from the attempt of using an existing DVE system to support collaboration using haptic devices.

1. Introduction

The human being, as a general rule, requires social interaction independently of the form it may take. Consequently, it is not surprising to witness the huge success of various online games ranging from First Player Shooters (FPS) to Massive Online Role Playing Games (MORPG), which continues to thrive [16] independently of the current economic recession.

The nature of these networked applications consists of the creation of a computer-generated world where people may interact with the environment itself and other users. Putting to one side the substantial monetary budget in content creation, one is left with a distributed application that is a rudimentary instantiation of a Distributed Virtual Environment (DVE).

The history of the DVE is recent and brief [28], albeit rapid evolution from the text-based environments; and Multi-User Dungeons (MUDs) to more sophisticated 3D immersive environments. With the computer being a mass-market commodity and the cheap network connectivity provided by the Internet, DVEs have abandoned the exclusivity of high technology research laboratories and become accessible by non-expert end-users at their homes. This trend continues to be re-enforced as consensus is established and standardization is gradually achieved, such as the case of eXtensible 3D (X3D) [31] that describes 3D content, namely geometry and some elements of behaviour.

Unfortunately, the existing DVE systems are fraught with technical difficulties that ultimately affect the user experience by disrupting both their sense of presence [3] and co-presence [23]. Most of these problems are rooted in the network support of the systems for effectively maintaining the illusion of a consistent environment that is distributed amongst geographically dispersed hosts. In fact, most of the DVE work is relatively successful within controlled conditions such as within a LAN, but are unsuccessful when deployed on the Internet demonstrating different failure thresholds. The transition to the Internet is not the culprit; rather it merely magnifies the existing network problems contained within systems that otherwise may go unnoticed.

Much of the problem may be found in the fact that DVEs have emerged as part of the natural transition from single-user to multi-user systems when integrating networking capabilities. The development emphasis goes towards the graphical technical issues since it affects the visual experience of the user. However, the current implementations and associated problems demonstrate a poor understanding of the network and its behaviour. A classical example is the preliminary transition of the Doom™ game from single player to multiplayer. The first approach was to send a network data packet per keystroke, quickly saturating the network with the increase in traffic. However, DVE are also prone to the same mistakes as the Networked Virtual Reality (NVR) system that [2] demonstrates. In this particular case, updates were sent every n^{th} frame, which could rapidly lead to network problems and overwhelm the world server since it was based on a client/server architecture.

This paper reports the failure of adopting an existing DVE system to support an Internet2 experiment across the Atlantic. The measurements collected from monitoring the network demonstrated that the source of the difficulties was not due to the network infrastructure, but the DVE system itself. The initial experiment design involved the usage of haptics, but this requirement made the problems even more evident. Ultimately, the inclusion of haptics in the experiment setup had to be abandoned. The lessons learnt from the experiment provided the necessary input for the design and implementation of a system to support collaborative manipulation of objects with haptics. Consequently this paper presents the initial prototype and the current development effort based on the data collected from a small study carried out.

The remainder of the paper is divided into an additional six sections, with the next indicating the common perception of the Internet by both the online game and VE developer communities. This is followed by a discussion of why haptic feedback is important and its impact on the sense of co-presence. In section four, the case study experiment is presented, along with the problems that compromised the initial design, requiring modifications. The experience garnered from this work has lead to design principles that are the basis of a prototype system. This system will be improved as discussed in the final section concerning conclusions and future work.

2. The Internet “Sucks”

The case study of [15] clearly denotes the prevalent opinion of the games development community towards the Internet. The related project resulted from the requirement to extend an existing 3D game to support multiple players and the author relates most of the pitfalls encountered, along with devised solutions.

The conclusion that the Internet “sucks” denotes a clear misunderstanding of how the network works beyond the controlled behaviour of a LAN. Therefore it is worthwhile to comment on the report since it represents common misconceptions throughout the design of most online games and DVE systems.

2.1 What is the Internet?

The underlying service model of the Internet is based on “best effort”, meaning there are no guarantees regarding its performance and consequently no optimal expectations should be part of the design of network subsystems.

Taking a simplified view of the Internet we can regard it as an enormous global graph where the nodes are network elements, namely hosts and routers. The latter is responsible for directing incoming data packets along the correct outgoing path. Since no fixed routes exist, the complete path from a source to destination is asymmetrical and may vary significantly from one data packet to another. At each router, the incoming packets are placed in queues before being directed to the next appropriate destination. However, as queues have finite dimensions regarding packet capacity, which may result in packets being occasionally dropped should the capacity be exceeded.

The Internet is a resource shared amongst millions of hosts, thus it is expected to experience problems that affect the following properties:

- **Bandwidth**. This denotes how many packets per unit of time may pass a particular path connecting two adjacent routers. The bandwidth may vary along the desired path.
- **Latency**. This denotes the time it takes for a packet to arrive at its destination.
- **Jitter**. This denotes the variance in the inter-arrival times of the data packets.
- **Loss**. This denotes the number of packets being lost.

Traditionally, these properties are measured at the hosts, giving an assessment of the expected overall network properties, as a well behaved black box, between the source and destination. However, it has been demonstrated that the traffic generated on the Internet is self-similar [8], making it difficult to employ statistical models, such as the Poisson distribution [19].

Although there are mechanisms for Quality of Service (QoS) to enhance the current model of the Internet [4,5,25,29,30], these are not widely deployed. In addition, the overhead involved may be detrimental to the interactivity requirements of the DVEs and when the network is stressed the service guarantees may fail as reported in [14]. In this particular case, Differentiated Services (DiffServ) [4] was used to provide guarantees of bandwidth and latency. However, heavy network load affected the established service parameters increasing the packet loss that ultimately failed to maintain the expected latency value although bandwidth was assured.

2.2 TCP vs UDP

The Internet Protocol (IP) stack isolates any application from the details pertaining the routing of data packets by providing the Transport layer. This layer provides end-to-end connectivity, allowing the system to focus on preparing data to be sent and on processing receiving data without regard to how the resulting packets are routed between sources and destinations.

The most common protocols of the transport layer, available to system developers, are the Transmission

Control Protocol (TCP) [22] and the User Datagram Protocol (UDP) [21].

- **TCP**. The system developer has a First In First Out (FIFO) unicast connection with total ordering and total reliability. The programming interface provides an abstraction of the network that corresponds to a stream where the all the data written has the guarantee of ordered delivery. The protocol handles packet loss with retransmission, taking care for any duplicates.

  TCP also takes into account that the network is a shared resource, thus if loss is experienced, it may imply that the queues along the path are congested. As a result the packet transmission rate is reduced until the state of the network is deemed to have recovered. The process is known as congestion control.

  To support these characteristics, an elaborate buffering mechanism coupled with a control protocol exists, which results in an overhead detrimental to the requirements of real-time interactivity.

- **UDP.** This option provides a connectionless service that may be either unicast or multicast. There is absolutely no guarantee of delivery itself or the order in which the packets arrive at the remote host. This transport protocol provides maximum flexibility at the expense of requiring development effort for implementing the necessary mechanisms to support the delivery requirements of the data.

  However, the majority of the Internet traffic remains TCP based due to the World Wide Web (WWW), thus there is a need for all remainder traffic to be TCP-friendly to avoid congestion collapse [10]. Therefore there should always exist some mechanism of congestion control, particularly if any form of reliability is required. Otherwise, once congestion is experienced, the TCP flows may reduce the rate to allow the network to recover but the situation will not improve if the UDP flows do not also stop sending data packets. Therefore, the argument supported by some developers that UDP is better than TCP because it does not perform congestion control may prove disastrous to the overall state of the Internet.

Although online games and other more traditional DVE systems adopt a single solution based on either UDP or TCP, in reality the choice is not simple. Unfortunately, in most cases, systems fail to recognise the complexity of the data involved to successfully support a DVE. As described in [6], the requirements of the data are widely varied and not necessarily compatible with one another, thus there is no single protocol that will fulfil all the needs of a DVE. Not even the widely known Real-time Transport Protocol (RTP) [24], which does not adequately support data with high variance though it is ideally suited for streaming media such as video and audio [20].

2.3 Architecture Design

The design choices made for the communication infrastructure are highly influential in the overall architecture of a system. Thus, the use of TCP traditionally leads towards Client/Server, whilst the adoption of UDP multicast results in distributed architectures.

With online games, the most prevalent architecture is Client/Server. The main reason for this design is the ease of maintaining the consistency of the DVE; the end-hosts are security threats being susceptible to cheating and secondly, the implementation is relatively simple. However, it is not possible to eliminate the inherent bottleneck at the server. This compromises scalability of the system beyond a certain threshold that is dependent of the available resources (network and computational) at the server.

Another alternative that exists in experimental DVE systems is the distributed architecture. The complexity regarding consistency increases significantly, but the scalability of the system is improved. Traditionally this approach relies on multicast, which is not widely available to the market by the various Internet Service Providers (ISP) due to deployment problems ranging from technical to business nature [9]. However, both the ISPs and research community are addressing the problems and solutions are gradually being devised. In the meantime, some systems have devised software solutions, which may not be totally satisfactory, but are functional [11].

2.4 Internet2

The Internet2 [32] is a research initiative involving universities, research institutes and companies involved in improving the backbone of the Internet and deploying applications that would benefit from the increased bandwidth, lower loss and latency.

In Table I shows the results from some traffic monitoring between University College London (UCL) and the two US remote sites, Massachusetts Institute of Technology (MIT) and University North Carolina (UNC). The measurements were obtained by the use of the network tool pchar.

Table 1 - Measurements between UCL and two US sites

	UCL - MIT	UCL – UNC
Hops	12	14
Loss	<0.1%	<0.1%
RTT	91.5 ms	88.8 ms

As seen, there is practically no data packet loss in average across the Atlantic. The Round-Trip-Time (RTT) is well within the suggested threshold of 150ms suggested by [18] to support interactivity in a DVE.

3. The "Sense" of Touch

With single user virtual environment systems, the objective is to guarantee the immersion of the user so they feel present in the alternate reality at all times. However, when considering DVE, the sense of presence is not the sole factor to consider since a user is sharing the environment with any number of other users. Thus, it is important to also guarantee the sense of co-presence. In fact, some studies [26] have demonstrated that co-presence contributes significantly to the sense of presence.

Most of the interaction modalities available in today's DVE are focused solely on visual and auditory senses. The use of haptic devices allows a user to experience forces, thus making it possible the sense of touch. The inclusion of haptic feedback increases the user's sense of presence.

In [1], an experiment was carried out to validate the impact of haptics on the sense of co-presence. The experiment consisted of two users manipulating a ring along a curved wire without touching it. If the ring did touch the wire, the colors of the screen would change until the error was corrected. Two conditions were explored:

- Only visual feedback;
- Visual and haptic feedback.

The task performance of the subjects increased significantly when haptic feedback was used instead of just visual. The results also indicate that the sense of co-presence is increased by inclusion of haptic feedback.

Although the study aimed to correlate haptics with co-presence, implicating the involvement of two users, the infrastructure setup was based on a single computer with two haptic devices and two monitors connected to it. Thus, the associated problems of involving a network were avoided.

The inclusion of a network between two remote hosts adds additional complexity to the problem domain, since the latency introduces synchronization inconsistencies. These problems are difficult to solve and generally magnify erroneous behavior in existing DVE systems.

In [7], the study was done using a LAN as the network infrastructure. So all the users interacted through separate computers each with a haptic device connected to it. In order to emulate the Internet, an artificial delay was introduced, but this is not sufficient as demonstrated in [15]. The behavioral analysis of the Internet cannot be reduced to just latency. It is necessary to consider the loss and jitter, along with data packets out of order due to the asymmetric nature of the Internet. Without considering these issues, once the system is deployed across the Internet, the results may be appalling even if the latency experienced is less than the one emulated on the LAN during development [15].

4. The Experiment

The initial objective of the experiment [17] was to study the use of haptics in a DVE across transatlantic links. The target DVE system to be used was Distributed Interactive Virtual Environment (DIVE)[1] [13] with minor modifications to integrate the haptic devices.

The choice of DVE system was based on six-years experience using it in collaborative virtual environment research. Some of the strong advantages of the system include its simplicity of use. The possibility of loading different geometry file formats coupled with an easy scripting engine based on tcl/tk, allows a researcher to promptly setup experiments with minimal system tampering.

Previous experience was based on collaborative scenarios where the interaction amongst users was limited to social interaction based on speech and visual cues [26, 27]. The inclusion of haptics came to reveal some serious limitations in the DIVE system design regarding the networking infrastructure.

The following subsections will provide an overview of the experiment design and describe the network infrastructure along with the problems encountered that limited the scope of the initial research objectives. The actual results of the experiment are reported elsewhere [17].

4.1 Experiment Design Overview

The experiment involved two users that would collaborate in a joint task. The objective was to carry a flat object with two handles having objects resting upon it, as illustrated in Fig. 1. For convenience, the object was described to the subjects as a stretcher to be jointly carried with another user.

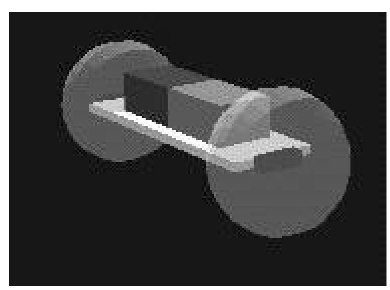

Fig. 1 - The "strecher" to be jointly carried by the users

The environment itself was reduced to bare essentials, thus providing minimal distraction to the task being carried out. The result, depicted in Fig.2, consisted of a

[1] The version used was 3.3

large building with a blue path starting outside and leading inside through a large entrance.

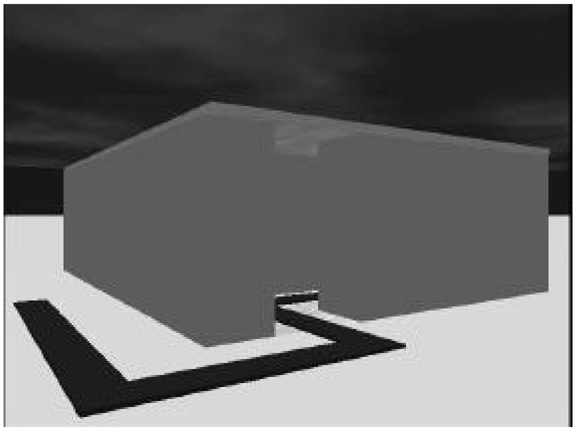

Fig. 2 - The environment with the simple building and the blue path indicating the trajectory to be followed

The path that subjects had to follow consisted of several axis-aligned segments. Although the subjects were encouraged to follow the path, they were briefed that the general direction of the path was more important than following it exactly.

The task would be initiated with the "stretcher" being located on the floor at the beginning of the path outside of the building. Successful completion of the task meant depositing the "stretcher" on a red square located within the building at the end of the blue path.

The study was conducted between UNC and UCL with a total of 17 subjects all recruited at UCL. The experimenter at UNC acted as a confederate throughout the experiment, aiding the subjects in their task. The assigned chore was terminated either upon successful completion or when the time limit of 8 minutes was reached.To assist a user in discerning the position and orientation of their remote partner, users were represented by the block avatar of Fig.3.

Fig. 3 - Visual representation of a user's avatar

The avatar could only move its head and the pointer indicated the position of the hand. These minimal cues, in terms of body language, were complemented by the use of audio communication to allow participants to verbally negotiate their progress along the designated path.

The hardware setup used at UCL consisted of a ReaCTor system, consisting of four projection walls each with an area of 3x2.2 meters. The subjects would control their navigation and manipulation via a joystick with four buttons. They would wear, in addition to the goggles, the Intersense tracking device. The setup at UNC is described in [33].

4.2 Network Infrastructure

A high level overview of the network topology used is depicted in Fig. 4. Although the actual experiment was conducted between UNC and UCL, initial testing and evaluation was carried out between all sites.

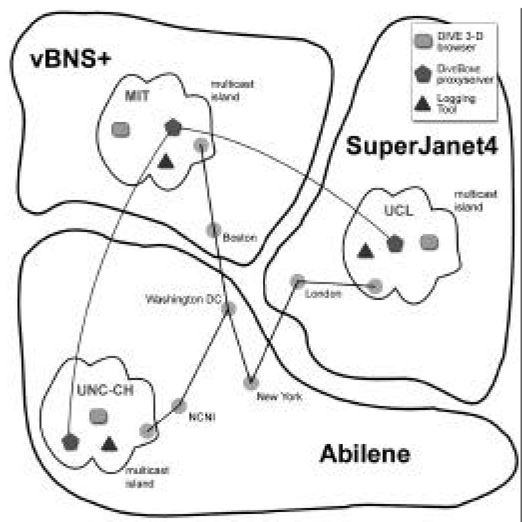

Fig. 4 - Network topology of the interconnecting sites

As previously mentioned in section 2.4, along with table I, the overall performance of the network is adequate for real-time interaction in DVEs. The overall RTT between UNC and UCL stabilized around 80-90ms.

4.3 Problems Encountered

Although network performance was deemed adequate for the experiment via network monitoring, the initial objective of using haptics was compromised.

An initial pilot study was carried out with a simplified scenario where two remote users would collaborate together to lift a box and maintain it above a certain threshold. Each user would be required to exert a reasonable force on opposite sides of the box in order to achieve the necessary leverage to lift it. The forces would be required to be at an angle to actually raise the box.

However, even though network conditions were reasonable, the system's performance was incredibly poor, making it unfeasible to attempt the completion of the task. The nature of the problem is the different frequency of the updates of parallel data structures as seen in Fig. 5.

The visual renderer handles the visual aspect of the DVE by performing the traversal of some data structure, typically an object scenegraph. To simulate the force feedback within a DVE, the haptic scenegraph is used to calculate all the forces applied to the objects and the reaction forces to be applied to the haptic device. Both the visual and haptic scenegraph are required to be consistent, so the feeling of touch reinforces what is visually perceived and vice-versa.

Now traditionally, the renderer performs frame updates between 10Hz to 60Hz depending on the performance of the computational resources (CPU, graphics card, etc). The haptic device on the other hand performs updates at 1KHz, which is of two orders of magnitude greater than the renderer. This implies different processing threads to handle the divergent frequency rates. However, in DIVE the event loop is tightly coupled to the rendering loop, thus events may only be processed at a speed that is either the same or less than the frame rate. With remote collaboration, the local haptic events are required to be processed not only locally, but also need to be sent to other participating hosts. With the low frequency rate coupled with the delay of the network, no matter how small, the visual and haptics scenegraphs would become unsynchronized quite quickly. Therefore, it was concluded that it would not be possible to use haptic devices with DIVE.

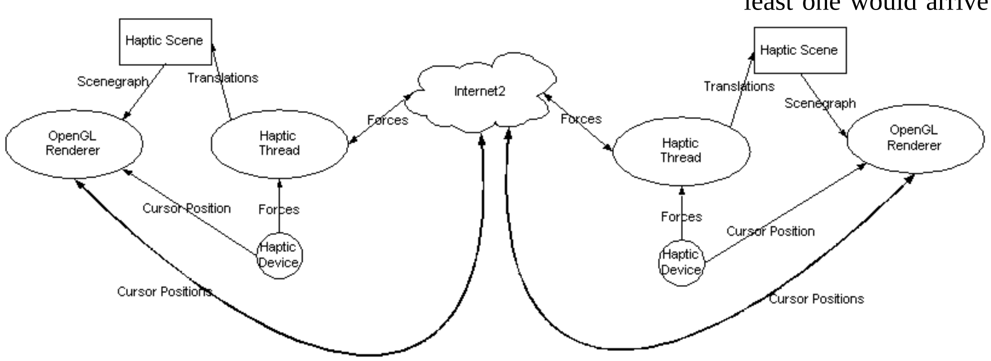

Fig. 5 - Overview of the parallel data structures within a DVE system with haptic device

The experiment carried out [17] was based on DIVE but without the inclusion of force-feedback, which limited the negotiation process to visual feedback and audio exchange between users. This incurs difficulties not normally found in the real-life manipulation of objects. The existence of gravity and the physical constraints of a similar object to the virtual "stretcher" would restrict movement of the participants. So for example, when one user pulls the object, the other user is pulled along being obliged to follow the movement. Without the use of haptics, the experience becomes quite unrealistic with higher degree of disruptions in both sense of presence and co-presence.

However, the fact that joint manipulation of the object was the core requirement of the experiment yielded significant problems to the DVE system.

In DIVE, dead reckoning is used to reduce the necessary sampling rate of the avatar's movement by leveraging the ephemeral nature of the updates. However, the setup at UNC required some modifications that ultimately raised technical difficulties for the system. At UNC subjects could not navigate the environment merely by physically walking around in their wide-tracked environment due to the large scale of the virtual model and the physical space constraints. It was necessary to add additional buttons to their handheld device to emulate moving forward/backwards. The buttons would control the start and end of locomotion at a given velocity, thus conferring the users with the means of traveling large distances without the need of physically walking. Unfortunately, these events required total reliability, which is not guaranteed by the supporting protocol, thus the loss of an event would yield inconsistencies between remote users.

The existence of mechanisms for total reliability is non-trivial and the traditional methods of acknowledging would not provide an adequate solution. The timeliness of an event is a constraint that would not allow the necessary negotiation mechanisms based on the use of timers and control packet exchange. As a result the solution adopted was simple; several copies of the event would be sent so at least one would arrive at the remote site, which discarded all unnecessary duplicates.

The joint manipulation of the same object, identified as a "stretcher", posed several difficulties. It involved manipulating the local copy of the object and letting the DIVE system propagate translational and rotational changes to other remote copies of that object on the network thereby creating a sense of shared ownership of the entity in question. This mode of manipulation would only guarantee a synchronized environment as long as the changes were applied in the same order in both instantiations of the environment. In turn this would only be possible if the events were generated, sent and processed at a higher resolution than the frequency of the manipulation of the object. If not, there would be disparate states of the object in its various instantiations, each of which would then send updates of its global position resulting in significant jitter of the object. Also, it would continually swap between the local perceived state and the one received in the packets from other instantiations of the VE. Before any given frame is rendered the state of the object would be determined either by the local or the remote state due to processing of remote packets.

Until pilot experiments were run, the experiment was carried locally and as the LAN provided less than 10ms turn around times the system did not present any problems. As soon as a link up with UNC was carried out, thus experiencing a RTT of 80ms, the result was the shared object would jitter. This was resolved by implementing an alternative approach that employed distinct local copies of the contents of the stretcher and shared handles, each owned by a single avatar. A local TCL script then updated the distinct local object based on the state of shared global objects. The stretcher would then align locally based on the position/orientation of the handles. In this set-up direct manipulation of a shared object was avoided. The VE appeared synchronized and visually correct, even though the two instantiations would differ slightly due to the lag in updating the positions of the handles. The stretcher would align according to the position of the hand of the subject locally and the position of the rendered hand of the remote avatar. So the alignment of the stretcher was based on the information available locally at the time of rendering.

5. An Alternative Solution

The initial objective of evaluating remote haptic collaboration was abandoned due technical insufficiencies of the DVE system. Even with only visual and auditory feedback, serious problems remained that occasionally affected the experience of the users.

In order to address the initial goal of the study, a dedicated application was developed, but with the focus being on addressing the problems that afflicted DIVE.

The initial prototype consisted of a scene illustrated in Fig.6 where the users would have to manipulate together a blue box (dark shaded cube). The task was to try collaboratively apply opposing forces such that it was possible to lift the cube above a particular threshold and maintain it for a predetermined time.

No visual feedback was available regarding a representation of the remote user, neither was there any auditory feedback. Only a representation of the cursor indicating the contact point of the local and the remote user was portrayed. These limitations make the negotiation process extremely difficult without the usage of haptic feedback. To ameliorate the difficulties experienced, an enhancement to the user interface was implemented to explicitly denote where the remote user would perceive the position of the box by painting a transparent pink cube (lighter shaded cube). This improved the negotiation process, as the users could understand better what the other participant was doing when their models became unsynchronized.

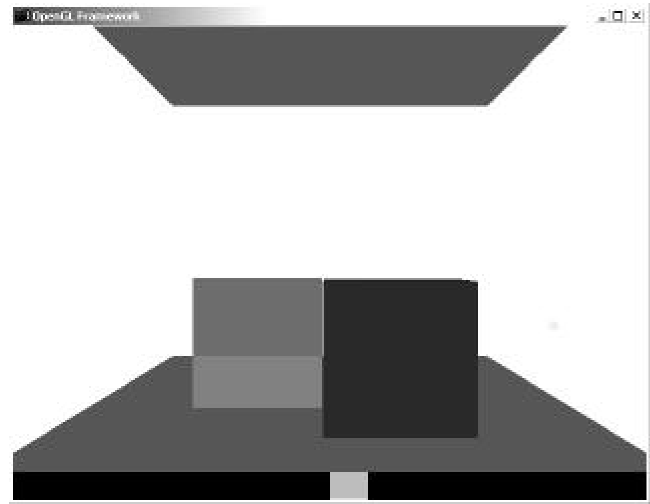

Fig. 6 - Screenshot of the user interface of the prototype system developed for collaborative haptics

Considering the limitations of coupling the event processing subsystem with the renderer subsystem, the system developed was based on a multi-threaded architecture. Therefore, it was possible for the rendering to be processed at 30Hz (or any other desired frame rate) whilst the haptic loop would run at 1KHz to fulfil the requirements of the haptic device. The network I/O code that related to the haptic subsystem was written into the haptic event loop.

Both disparate machines ran the code independently, and had their own copy of the environment. The system had inherent peer-peer architecture. This meant that any environmental change made at a machine was communicated and then applied at the remote instance.

The most effective method to connect the two distant instances of the environment was to send the reaction forces applied to the local cube to the remote system. Conversely, each instance applied any forces received over the network to its cube in addition to any forces applied by the local user.

The network subsystem was implemented with two optional transport protocols, one based on TCP and another on UDP. The latter required a header definition to uniquely identify the packet by including a sequence number and timestamp.

Empirically it was found that TCP was inadequate to support user interaction successfully. This was due to the sensitivity in terms of the latency, loss and jitter regarding the haptic device. With the relatively high frequency rate, the TCP sliding window (waiting on positive acknowledgements for every packet) and Addictive Increase Multiple Decrease (AIMD) behavior of the congestion control would wreak havoc in any attempt of collaboration.

The protocol based on UDP presented better results since a design choice was not to implement a total reliable transport. The main two reasons were the low loss experienced on the network, along with the timeliness requirements of the data that would render pointless any retransmissions due to exceedingly large delays. If a loss did occur, it would affect the environment by reducing the power of a force, and therefore desynchronize the location of the cube instances in the shared environment. Although little packet loss was experienced, there would be considerable jitter at times that would lead to temporary inconsistencies with abrupt feedback forces.

A small study was carried out [12] to evaluate the prototype and the results are positive, with haptic feedback playing a major role in reinforcing the sense of co-presence. Based on the data collected from both the application and network monitoring, further refinements are underway to improve the system.

6. Conclusions and Future Work

Both the games industry and the DVE research community view the network in an overly simplistic fashion, reduced to sending and receiving messages via different types of sockets.

This paper reported on the experience garnered during the study of collaborative haptics in trans-Atlantic experiments. The Internet-2 revealed to have adequate performance to support DVE interaction, but the strict requirements of haptic collaboration exposed pitfalls in an existing DVE system. The problems identified are not exclusive to the system used, but common in existing DVE systems and online games.

Ultimately, the initial objectives of the experiment were modified and the lessons learnt have lead to the development of a DVE system. The preliminary results of the system's usage are promising, but further work is required to improve the performance and haptic interaction. The next phase of the work will include a more sophisticated buffering mechanism coupled with an adequate physical model to provide further synchronization and smoothing.

7. Acknowledgements

This work was partially funded by the EPSRC EQUATOR project and the UKERNA project.

8. References

1. Basdogam, C., Ho, C., Srinivasan, M., and Slater, M., "An Experimental Study on the Role of Touch in Shared Virtual Environments", ACM Transactions on Computer-Human Interaction, Vol. 7, N. 4, December, 2000
2. Berger, J., Dinh, L., Masiello, M. and Schell, J., "NVR: A System for Networked Virtual Reality", Proc. IEEE something, 1994
3. Biocca, F., "The Cyborg's Dilemma: Progressive Embodiment in Virtual Environments ", Journal of Computer Mediated Communication, Vol. 3, N. 2, September 1997
4. Blake, R., Black, D., Carlson, M., Davies, E., Wang, Z. and Weiss, W., "An Architecture for Differentiated Services", IETF RFC 2475, December 1998
5. Braden, R., Clark, D. and Shenker, S., "Integrated Services in the Internet Architecture: An Overview", IETF RFC, July 1994
6. Brutzman, D., Zyda, M., Watsen, K. and Macedonia, M., "Virtual Reality Transfer Protocol (vrtp) Design Rationale", Workshops on Enabling Technology: Infrastructure for Collaborative Enterprises (WET ICE): Sharing a Distributed Virtual Reality, MIT, Cambridge Massachusetts, June, 1997
7. Choi, H., Choi, B. and Ryew, S., "Haptic Display in the Virtual Collaborative Workspace Shared by Multiple Users", Proc. IEEE International Workshop on Robot and Human Communication, 1997
8. Crovella, M. and Bestavros, A., "Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes", IEEE/ACM Transactions on Networking, Vol. 5, N. 6, December, 1997
9. Diot, C., Levine, B., Lyles, B., Kassem, H. and Balensiefen, D., "Deployment Issues for the IP Multicast Service and Architecture". IEEE Network magazine special issue on Multicasting. January/February 2000
10. Floyd, S. and Fall, K., "Promoting the Use of End-to-End Congestion Control in the Internet", IEEE/ACM Transactions On Networking, Vol 7, N. 4, August 1999
11. Frecon, E., Greenhalgh, C. and Stenius, M., "The DiveBone – An Application-Level Network Architecture for Internet-Based CVEs", Proc. ACM VRST'99, London, 1999
12. Jordan, J., Mortensen, J., Slater, M. and Oliveira, M., "Collaboration in a Mediated Haptic Environment", Proc. Presence 2002: 5th International workshop on Presence, Porto, 2002 (submitted)
13. Hagsand, O., "Interactive Multiuser VEs in the DIVE System", IEEE Multimedia, spring, 1996
14. Leigh, J., Yu, O., Schonfeld, D., Ansari, R., He, E., Nayak, A., Ge, J., Krishnaprasad, N., Park, K., Cho, Y., Hu, L., Fang, R., Verlo, A., Winkler, L. and DeFanti, T., "Adaptive Networking for Tele-Immersion",
15. Lincroft, P, "The Internet Sucks: Or, What I Learned Coding X-Wing vs TIE Fighter", Proc. Game Developer\s Conference, 1999
16. Markoff, J., "Recession? Don't Tell the Video Game Industry", The New York Times, 24th May, 2002
17. Mortensen, J., Vinayagmoorthy, V., Slater, M and Steed, A., "Collaboration in Tele-Immersive Environments", Proc. 8th Eurographics Workshop on Virtual Environments, Barcelona, June, 2002

18. Park, K. and Kenyon, R., “Effects of Network Characteristics on Human Performance in a Collaborative Virtual Environment”, Proc. IEEE VR’99, Houston, March, 1999
19. Paxson, V. and Floyd, S., “Wide-Area Traffic: The Failure of Poisson Modeling”, Proc. ACM SIGCOMM’94, London, 1994
20. Perkins, C. and Crowcroft, J., “Notes on the use of RTP for shared workspace applications”, ACM Computer Communication Review, Vol. 30, N.2, 2000
21. Postel, J., “User Datagram Protocol”, RFC 768, 1980
22. Postel, J., “Transmission Control Protocol”, RFC 793, 1981
23. Schroeder, R., Steed, A., Axelsson, A., Heldal, I., Abelin, A., Wideström, J., Nilsson, A. and Slater, M., “Collaborating in networked immersive spaces: as good as being there together?”, Computer & Graphics, Elsevier Science Ltd, Vol. 25, N. 5, October 2001
24. Schulzrinne, D., Casner, S, Frederick, R. and Jacobson, V., “RTP: A Transport Protocol for Real-Time Applications”, RFC 1889, 1996
25. Shenker, S., Partridge, C and Guerin, R., “Specification of Guaranteed Quality of Service”, IETF RFC 2212, 1997
26. Slater, M., Sadagic, A., Usoh, M. and Schroeder, R., “Small Group Behaviour in a Virtual and Real Environment”, Presence: Teleoperators and Virtual Environments, Vol. 9, N. 1, 2000
27. Slater, M., Howell, J., Steed, A., Pertaub, D. and Garau, M., “Acting in Virtual Reality”, ACM Collaborative Virtual Environments, September, 2000
28. Waters, R. and Barrus, J., “The Rise of Shared Virtual Environments”, IEEE Spectrum March 1997
29. White, P., “RSVP and Integrated Services in the Internet: A Tutorial”, IEEE Communications Magazine, May 1997
30. Wroclawski, J., “Specification of the Controlled Load Network Element Service”, IETF RFC 2211, 1997
31. www.web3d.org
32. Yeung, F., “Internet 2: Scaling Up the Backbone for R&D”, IEEE Internet Computing, Vol. 1, N.2, 1997
33. Welch, G., Bishop, G., Vicci, L., Brumback, S. and Keller, K., “The HiBall Tracker: High Performance Wide Area Tracking for Virtual and Augmented Environments”, ACM VRST’99, December 1999